PASSED

732A38: Computer lab 5
Computational statistics

Caroline Svahn & Martina Sandberg

March 9, 2016

1 Hypothesis testing

In 1970, the US Congress instituted a random selection process for the military draft. All 366 possible birth dates were placed in plastic capsules in a rotating drum and were selected one by one. The first date drawn from the drum received draft number one, the second date drawn received draft number two, etc. Then, eligible men were drafted in the order given by the draft number of their birth date. In a truly random lottery there should be no relationship between the date and the draft number. Your task is to investigate whether or not the draft numbers were randomly selected. The draft numbers (Y=Draft_No) sorted by day of year (X=Day_of_year) are given in the file **lottery.xls**.

1.1

Make a scatterplot of Y versus X and conclude whether the lottery looks random.

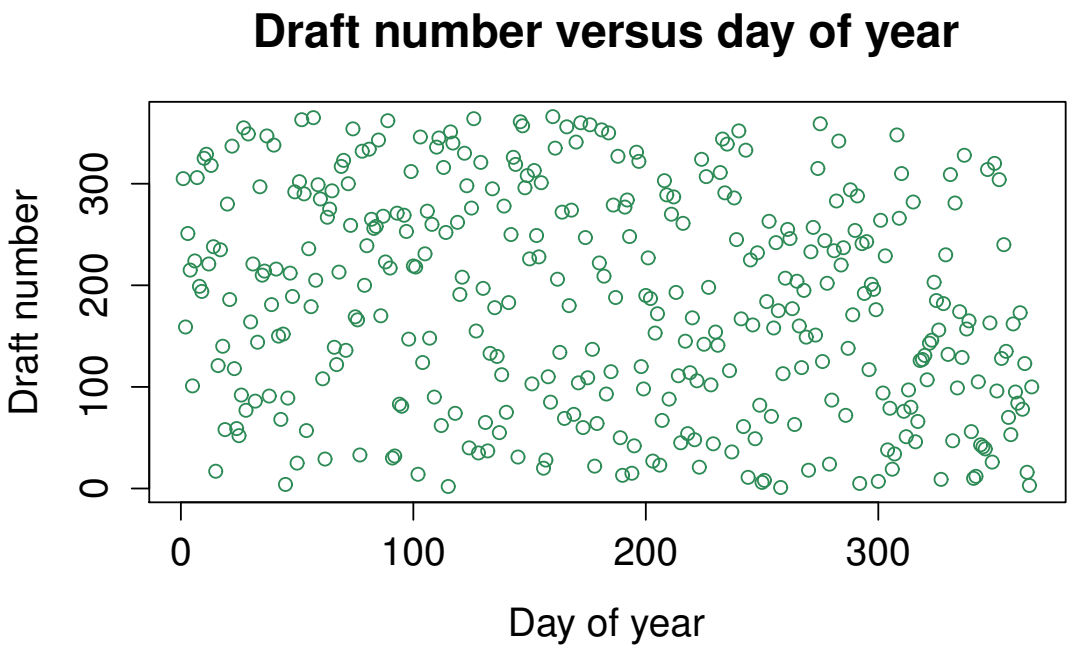

Figure 1: Scatterplot of draft number versus day of year.

Figure 1 shows a scatterplot of the draft number against the day of the year. From this plot, one may conclude the lottery to be random. This since the observations are well scattered without any

evident clusters.

1.2

Compute an estimate $\hat{Y}$ of the expected response as a function of X by using a loess smoother (use `loess()`*), put the curve $\hat{Y}$ versus X in the previous graph and state again whether the lottery looks random.*

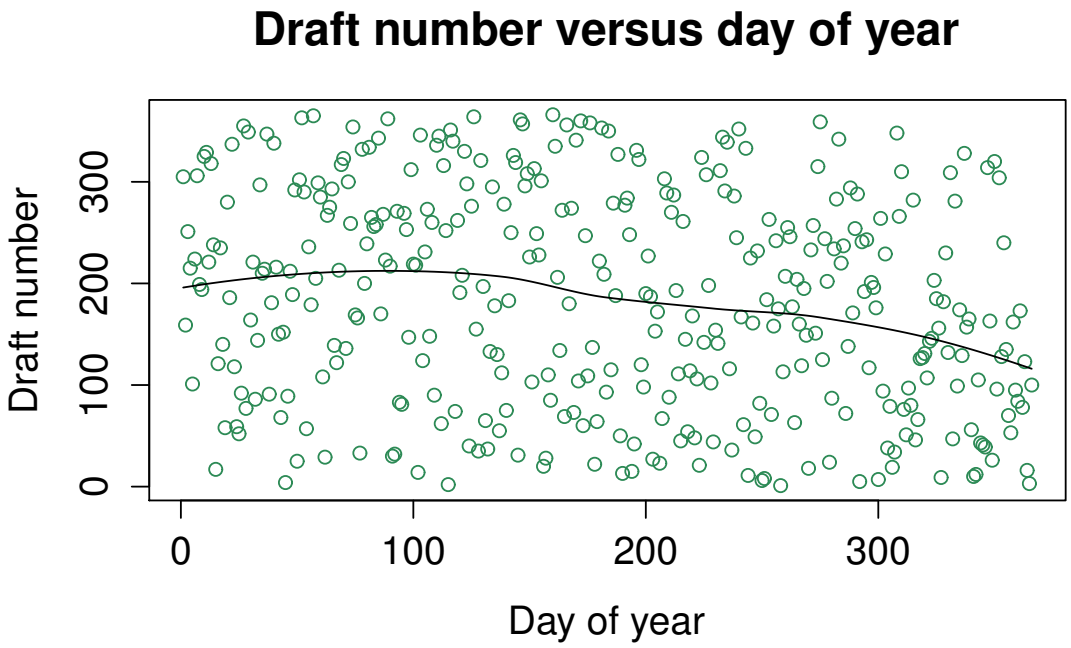

Figure 2: Scatterplot of draft number versus day of year with $\hat{Y}(x)$ as the line.

If the lottery is indeed random, then the slope of the estimated function should be zero and located at the mean. Figure 2 shows the same scatterplot as in 1.1 but here we also have $\tilde{Y}$ as the black line. The level of the estimated function is not quite constant. After the 100 first days of the year, the draft number decays somewhat. From this we can conclude that the lottery may not be random after all.

1.3

To check whether the lottery is random, it is reasonable to use test statistics

$$T = \frac{\hat{Y}(X_b) - \hat{Y}(X_a)}{X_b - X_a}$$

where $X_b = argmax_x \hat{Y}$, $X_a = argmin_x \hat{Y}$. If this value is significantly greater than zero, then there should be a trend in the data and the lottery is not random. Estimate the distribution of T by using a non-parametric bootstrap with $B = 2000$ and comment whether the lottery is random or not. What is the P-value of the test?

When estimating the distribution with non-parametric bootstrap, one estimates $T(D)$, the desired estimate, from the original data. Then, a new data set D_1 with the same dimensions is generated

by sampling with replacement from D. The procedure is then repeated B times, always using the data set D for sampling. The distribution is then given by $T(D_1) \dots T(D_B)$

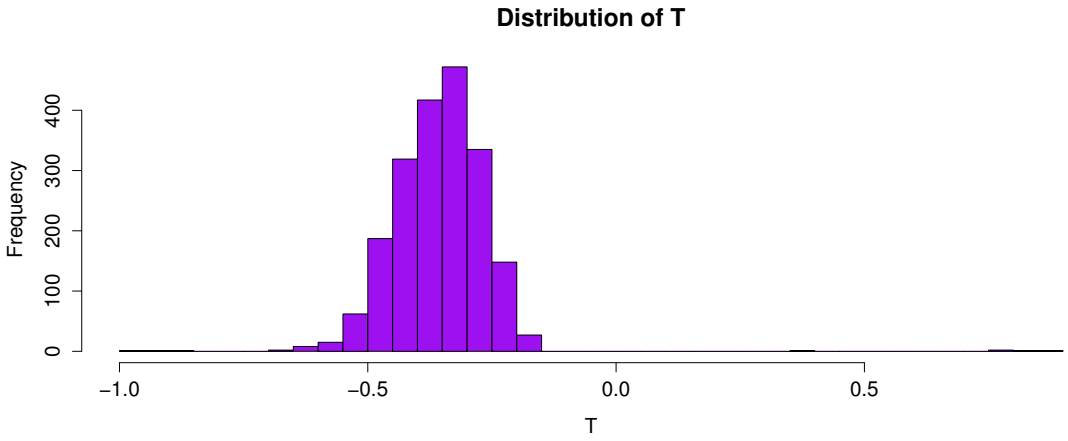

Figure 3: Histogram of the T-statistics generated from the bootstrap.

In this case, we have the null hypothesis $H_0 : T = 0$, that is that the data is random because there is no trend in the data. The alternative hypothesis $H_0 : T \neq 0$ states that the data is not random because then there is trend in the data. In figure 3 we can see how the bootstrapped T:s are distributed. Here we can clearly see that T is not likely to be zero. To find the p-value for the null hypothesis, each value of T is compared to 0. If the value of T_i is greater than 0, the result will be 1, otherwise 0. The p-value is then found by computing the mean of these boolean values. With seed 12345 and $B = 2000$, the obtained p-value is 0.0025. Because this value is smaller than 0.05, there is a very small chance that T is greater than 0, and thus we can reject the null hypothesis and conclude that there is no evidence of the lottery sample being random.

1.4

Implement a function depending on data and B and that tests the hypothesis H_0: Lottery is random vs H_a: Lottery is not random by using a permutation test with statistics T and returns the p-value of this test. Test this function on our data with $B = 2000$.

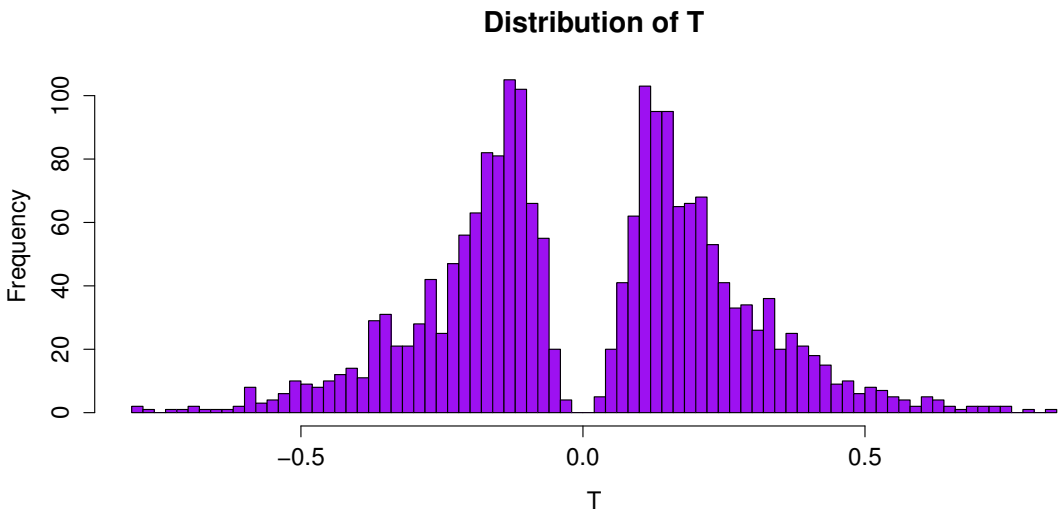

Figure 4: Histogram of the T-statistics generated from the permutation test.

If H_0 is not true we would expect to observe larger values of $|T_0|$ than if H_0 were true. The further away from 0 the value of T_0 we observe, the stronger is the evidence against H_0. With the permutation test we generate some T-statistics that shows us how the distribution of them looks like under H_0. If our T_0 is within the 95% confidence zone of this distribution we can not reject H_0, and otherwise we do. We get p-value = 0.159, which shows that our T_0 is not significantly different from the other statistics. This p-value is greater than 0.05 so, in this case, we do not reject the null hypothesis. The data may be random.

1.5

Make a crude estimate of the power of the test constructed in step 4:

a) Generate (an obviously non-random) dataset with $n = 366$ *observations by using same* X *as in the original data set and* $Y(x) = max(0, min(\alpha x + \beta, 366))$ *where* $\alpha = 0.1$ *and* $\beta \sim N(183, sd = 10)$.
b) Plug these data into the permutation test with $B = 200$ *and note whether it was rejected.*
c) Repeat steps a)-b) for $\alpha = 0.2, 0.3, .., 10$.

What can you say about the quality of your test statistics considering the value of the power?

To compute the power of the test statistics we generate data samples that satisfy H_a, which is done in a). Now the power is the percent of correct rejections. Starting out with $\alpha = 0.1$, we obtain a p-value of 0, concluding the sample to be non-random. Repeating the process for all values also obtains 0, for all values of α. This is expected since the Y-values used are not random, no matter the value of α. Because all p-values became 0 we have 100% correct rejections, thus our test statistics are good.

2 Bootstrap, jackknife and confidence intervals

The data you are going to continue analyzing is the database of home prices in Albuquerque, 1993. The variables present are: Price; SqFt - the area of a house; FEATS - number of features such as dishwasher, refrigerator and so on; Taxes - annual taxes paid for the house. Explore the file **prices1.xls**.

2.1

Plot the histogram of Price. Does it remind any conventional distribution? Compute the mean price.

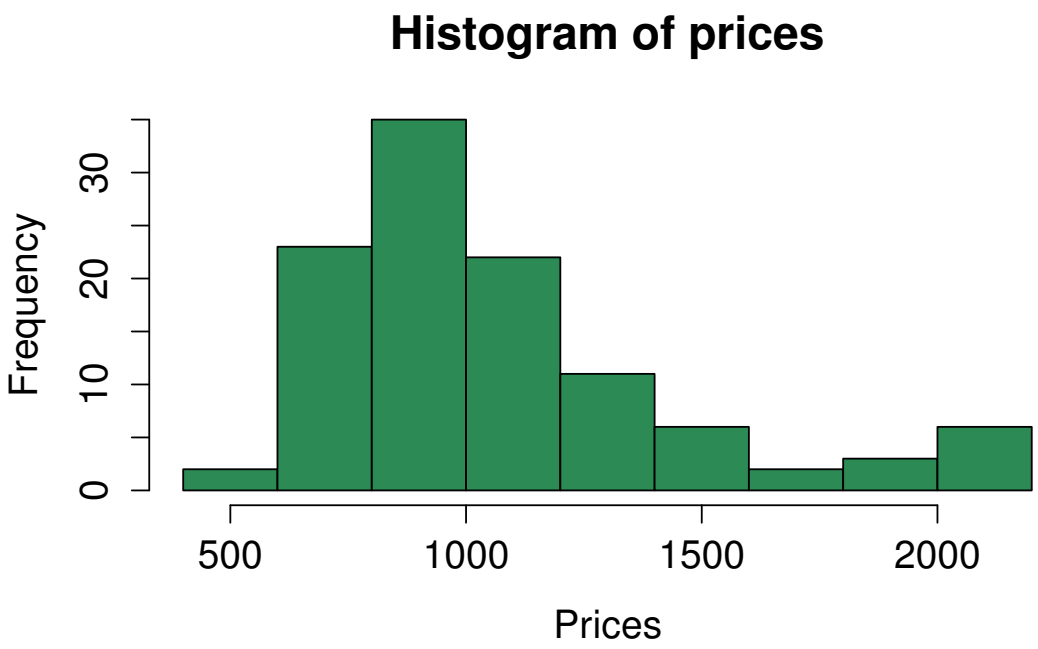

Figure 5: Histogram of the Prices.

The distribution of the price data is visualized in figure 5. From this, the variable might be either beta distributed or normal - since the histogram is skewed, it resembles a beta distribution with a high value of α and a small value of β, however, since n is rather small, the distribution might still be normal. It might also be the case that the prices are not distributed as any conventional distribution. The mean price is 1080.47.

2.2

*Estimate the distribution of the mean price of the house using bootstrap. Determine the bootstrap bias-correction and the variance of the mean price. Compute the 95% confidence interval for the mean price using bootstrap percentile, bootstrap BCa, and first-order normal approximation (**Hint**: use `boot()`,`boot.ci()`,`plot.boot()`,`print.bootci()`)*

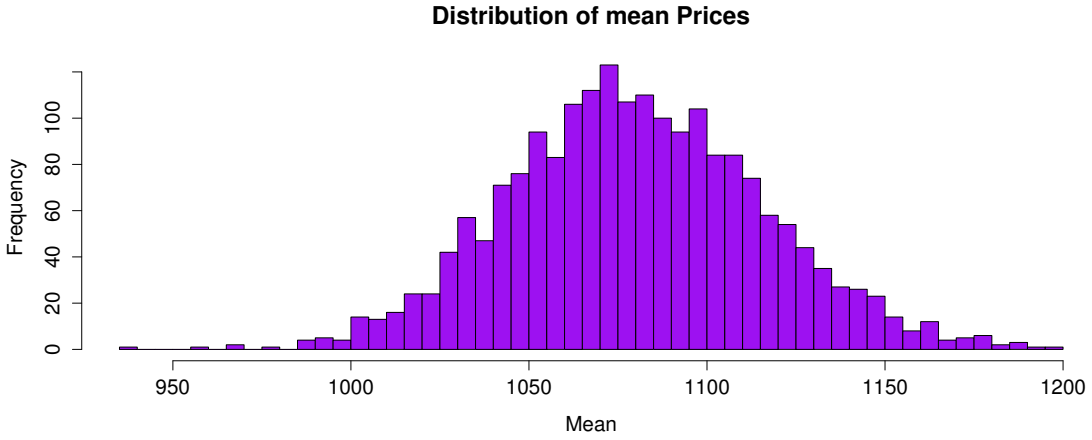

Figure 6: Histogram of the mean price of the housing generated from bootstrap.

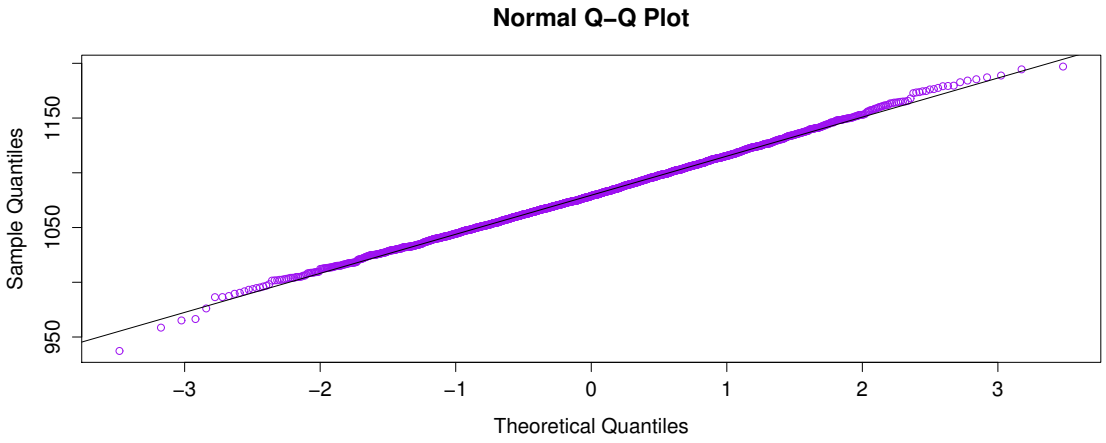

Figure 7: Normal plot of the mean price of the housing generated from bootstrap.

In figure 6 the mean prices from the bootstrap (with R=2000) is plotted in an histogram. These mean prices looks normal distributed, which can be confirmed by the normal Q-Q plot in figure 7, which also look normal. The bootstrap bias-correction can be computed as two times the statistics with the original data minus the mean of the statistics we got from the bootstrap. When computed we get the bootstrap bias-correction 1080.96, and the variance of the mean price is $35.84^2 = 1284.506.$ The 95% confidence intervals for the mean price using bootstrap percentile, bootstrap BCa, and first-order normal approximation can be seen in table 1.

Method	**Lower**	**Upper**	**Length**	**Mean**
Percentile	1013	1152	140	1082.5
BCa	1018	1162	145	1090
Normal	1011	1151	141	1081

Table 1: 95% confidence intervals for the mean price.

2.3

Estimate the variance of the mean price using the jackknife and compare it with the bootstrap estimate.

$$\widehat{Var(T)} = \frac{1}{n(n-1)} \sum_{j=1}^{n} (T_j^* - \bar{T}^*)^2 \tag{1}$$

We obtain $\boldsymbol{Y}_{(-j)}$ by dropping one observation at the time, that is $j = 1, 2, \ldots, n$, where n is the total number of price observations in the data, so in this case $n = 110$. For each j we compute $T_{(-j)} = T(\boldsymbol{Y}_{(-j)})$. To compute the variance we use formula (1) where $T_j^* = nT - (n-1)T_{(-j)}$ and $\bar{T}^*$ is the mean of T_j^*. We get that $\widehat{Var(T)} = 1320.91$. In the jackknife case the variance is often overestimated, wich we also can conclude is the case if we compare this variance with the one we got in the bootstrap case which was 1284.51.

2.4

Compare the confidence intervals obtained with respect to their length and the location of the estimated mean in these intervals.

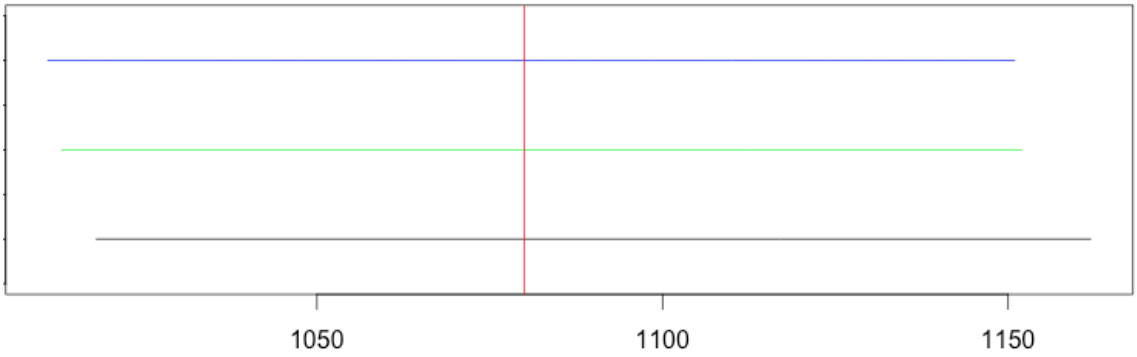

Figure 8: Confidence intervals. The red line is the estimated mean 1079.98, blue is the first order Normal approximation c.i, green is percentile c.i and black is the BCa c.i.

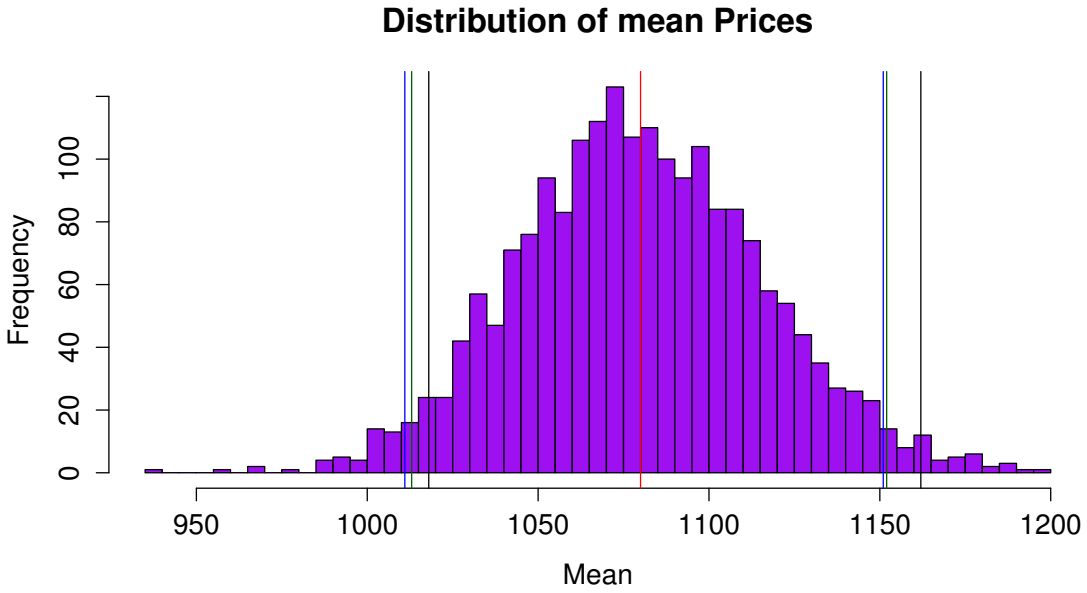

Figure 9: Histogram of the mean price of housing generated with bootstrap and confidence intervals. The red line is the estimated mean 1079.98, blue is the first order Normal approximation c.i, green is percentile c.i and black is the BCa c.i.

The mean of the bootstrap sampled T:s is 1079.98. In figure 8 the three confidence intervals are plotted. Here the red line corresponds to the estimated mean 1079.98, the blue line to the first order Normal approximation c.i, the green line to the percentile c.i and the black line to the BCa c.i. The normal and the percentile is very similar, while the BCa is a little more shifted to the right. All intervals have roughly the same length.The normal interval is a bit lower than the boot percentile interval and the BCa is located a bit higher than the boot percentile. The mean of the normal c.i is closest to the estimated mean, while the mean of the BCa c.i is fathers away. All intervals does however include the estimated mean with substantial marginal. The confidence intervals can also be seen plotted together with the mean histogram in figure 9.

Appendix

Contribution

About half of the plots are from Martina's report and the rest from Caroline's. The descriptions of the algorithms and interpretations are merged from both reports. The code is Martina's but Caroline's code is similar and yields basically the same results. Only one students code is used so that the results should be consistent for the tasks.

Code

```
## Assignment 1
X <- lottery$Day_of_year
Y <- lottery$Draft_No
# Dataframe with X and Y
dataframe <- data.frame(X=lottery$Day_of_year,Y=lottery$Draft_No)

# 1.
#(Y=Draft_No) sorted by day of year (X=Day_of_year)
plot(x=X,y=Y,pch=16,col="purple",xlab="Day␣of␣year",ylab="Draft␣number",main="Sc
```

```
# 2.
Yest <- loess(Y~X,data=dataframe)
Ytilde <- fitted(Yest)

plot(x=X,y=Y,pch=16,col="purple",xlab="Day␣of␣year",ylab="Draft␣number",main="Sc
lines(x=X,y=fitted(Yest),col="red")

# 3.
library(boot)

func <- function(data,i){
  data1 <- data[i,]
  Yest <- loess(Y~X,data=data1)
  Ytilde <- fitted(Yest)
  max <- match(max(Ytilde),Ytilde)
  min <- match(min(Ytilde),Ytilde)
  Xb <- data1$X[max] # find the x that maximizes Y
  Xa <- data1$X[min] # find the x that minimizes Y
  Tstat <- (max(Ytilde)-min(Ytilde))/(Xb-Xa)   # T
  return(Tstat)
}
set.seed(12345)
res <- boot(dataframe,func,R=2000)
plot(res)
hist(res$t,50,xlab="T",main="Distribution␣of␣T",col="purple")

mean(res$t>0) # p-value = 0.0025

# 4.
func2 <- function(data,B){
  Yest <- loess(Y~X,data=data)
  Ytilde <- fitted(Yest)
  max <- match(max(Ytilde),Ytilde)
  min <- match(min(Ytilde),Ytilde)
  Xb <- data$X[max] # find the x that maximizes Y
  Xa <- data$X[min] # find the x that minimizes Y
  Tstat0 <- (max(Ytilde)-min(Ytilde))/(Xb-Xa)   # T
  stat <- numeric(B)
  n <- dim(data)[1]
  for(b in 1:B){
    data$Y <- sample(data$Y, n)
    Yest <- loess(Y~X,data=data)
    Ytilde <- fitted(Yest)
    max <- match(max(Ytilde),Ytilde)
    min <- match(min(Ytilde),Ytilde)
    Xb <- data$X[max]
    Xa <- data$X[min]
    stat[b] <- (max(Ytilde)-min(Ytilde))/(Xb-Xa)
  }
  return(list(stat=stat,pval=mean(abs(stat)>abs(Tstat0))))
}
set.seed(12345)
fun <- func2(dataframe,2000)
stat <- fun$stat # Generated T:s
pval <- fun$pval # p-value = 0.159

hist(stat,100,xlab="T",main="Distribution␣of␣T",col="purple")
```

```
# 5.
# a
n <- dim(dataframe)[1]
a <- 0.1
Y <- numeric()
for (i in 1:length(X)){
  b <- rnorm(1,mean=183,sd=10)
  e <- min(a*X[i]+b,366)
  Y[i] <- max(0,e)
}
data2 <- data.frame(X,Y)
# b
set.seed(12345)
fu <- func2(data2,200)
fu$pval # p-value = 0


# c
pv <- NULL
for(j in seq(0.1,10,by=0.1)){
  n <- dim(dataframe)[1]
  a <- j
  Y <- numeric()
  for (i in 1:length(X)){
    b <- rnorm(1,mean=183,sd=10)
    e <- min(a*X[i]+b,366)
    Y[i] <- max(0,e)
  }
  data3 <- data.frame(X,Y)
  set.seed(12345)
  f <- func2(data3,200)
  f <- f$pval
  pv <- append(pv,f)
}


## Assignment 2
# 1
hist(prices1$Price,main="Distribution␣of␣Prices",xlab="Price",col="purple")
mean(prices1$Price) # 1080.473

# 2
#Estimate the distribution of the mean price of the house using bootstrap
func3 <- function(data,i){
  data1 <- data[i,]
  Ts <- mean(data1$Price)
  return(Ts)
}
set.seed(12345)
booty <- boot(prices1,func3,2000)
hist(booty$t,50,main="Distribution␣of␣mean␣Prices",xlab="Mean",col="purple")
plot(booty)

qqnorm(booty$t, main = "Normal␣Q-Q␣Plot",xlab = "Theoretical␣Quantiles", ylab =
qqline(booty$t)
```

```
biascorr <- 2*mean(prices1$Price)-mean(booty$t) # bootstrap bias-correction
#(sum((booty$t-mean(booty$t))^2))/(length(booty$t)-1)


# Compute the 95% confidence interval for the mean price
bo <- boot.ci(booty,conf = 0.95, type = c("perc", "bca","norm"))
#Intervals :
#Level      Normal              Percentile            BCa
#95%   (1011, 1151 )   (1013, 1152 )   (1018, 1162 )


# 3
#Y random sample
#T(Y) mean
func3 <- function(data){
  Ts <- numeric()
  n <- dim(data)[1]
  for(j in 1:n){
    data1 <- data[-j,]
    Ts[j] <- mean(data1$Price)
  }
  return(Ts)
}
Tj <- func3(prices1)
Tt <- mean(prices1$Price)
r <- dim(prices1)[1]
Tpunkt <- mean(Tj) #sum(Tj)/r
Tstar <- r*Tt-(r-1)*Tj
Tbar <- mean(Tstar) #sum(Tstar)/r
varT <- sum((Tstar-Tbar)^2)/(r*(r-1))

# 4
hist(booty$t,50,main="Distribution␣of␣mean␣Prices",xlab="Mean",col="purple")
abline(v=1079.98,col="red")
abline(v=1011,col="blue")
abline(v=1151,col="blue")
abline(v=1013,col="darkgreen")
abline(v=1152,col="darkgreen")
abline(v=1018,col="black")
abline(v=1162,col="black")

plot(x=seq(1011,1162,by=1),y=rep(0,152),type="l",col="white",ylim=c(0.3,0.9))
lines(x=seq(1011,1151,by=1),y=rep(0.8,141),type="l",col="blue")
lines(x=seq(1013,1152,by=1),y=rep(0.6,140),type="l",col="green")
lines(x=seq(1018,1162,by=1),y=rep(0.4,145),type="l",col="black")
abline(v=1079.98,col="red")
```